A Protocol to Detect Local Affinities Involved in Proteins Distant Interactions

Christophe Magnan, Cécile Capponi, François Denis

A Protocol to Detect Local Affinities Involved in Proteins Distant Interactions

Christophe N. Magnan Cécile Capponi
François Denis
Laboratoire d'Informatique Fondamentale de Marseille, Université de Provence
UMR 6166 CNRS, 39, rue F. Joliot Curie, F-13453 Marseille cedex 13, France
{Firstname.Lastname}@lif.univ-mrs.fr

Abstract

The tridimensional structure of a protein is constrained or stabilized by some local interactions between distant residues of the protein, such as disulfide bonds, electrostatic interactions, hydrogen links, Wan Der Waals forces, etc. The correct prediction of such contacts should be an important step towards the whole challenge of tridimensional structure prediction. The in silico *prediction of the disulfide connectivity has been widely studied: most results were based on few amino-acids around bonded and non-bonded cysteines, which we call* local environments of bonded residues. *In order to evaluate the impact of such local information onto residue pairing, we propose a machine learning based protocol, independent from the type of contact, to detect affinities between local environments which would contribute to residues pairing. This protocol requires that learning methods are able to learn from examples corrupted by class-conditional classification noise. To this end, we propose an adapted version of the perceptron algorithm. Finally, we experiment our protocol with this algorithm on proteins that feature disulfide or salt bridges. The results show that local environments contribute to the formation of salt bridges. As a by-product, these results prove the relevance of our protocol. However, results on disulfide bridges are not significantly positive. There can be two explanations: the class of linear functions used by the perceptron algorithm is not enough expressive to detect this information, or cysteines local environments do not contribute significantly to residues pairing.*

1. Introduction

Since many years, biology provides computer science with crucial problems. In particular, the prediction of proteins tridimensional (3D) structure starting from primary amino acids sequence is a current challenge for both biologists and computer-scientists. The 3D structure of proteins almost determines their functions. About 5 millions of proteins primary sequences are available in different databases, whereas approximatively only 40.000 3D structures are known. Moreover, determining 3D structures experimentally is a long, expensive and unreliable task. It is the reason why many researchers work at developing automatic learning methods for predicting the structure of new proteins from known, experimentally designed structures.

Different bioinformatics ways to determine proteins structures have been proposed. The most widespread method consists in predicting different structural elements and long-range contacts, then to propose the set of possible 3D structures matching these predictions. The prediction of the secondary (2D) structure, an intermediary structure between the amino-acids sequence and the 3D structure, have received considerable attention from researchers. β-sheets prediction [17, 21, 28, 26, 8] or α-helix [31, 26, 25] are examples of secondary structure elements.

Apart 2D structures, some punctual interactions between distant residues of the primary sequence of a protein, are also of interest, even if they are sometimes considered as a consequence of the 3D conformation rather than a cause. Among them, *disulfide bridges* have been widely studied. The prediction of such covalent bonds from primary sequences is a two-stages process: (i) prediction of the oxidization state of cysteines; (ii) prediction of the disulfide connectivity: which cysteine is bonded with such other given oxidized cysteine?

The first stage has been widely studied [13, 15, 22, 20, 19, 16, 6, 27, 7, 9], leading to fairly good performances: about 90% of correct predictions. Meantime, some methods have been proposed for the prediction of the correct connectivity [11, 12, 29, 30, 14, 32, 2, 7, 9]. However, none have reached 63% of correct connectivity despite their strong biological, theoretical, and algorithmic foundations. Moreover, the prediction results are non stable. Actually, most of these methods use the local environment around cysteines to predict their pairing (*i.e.* the 5 ou 6 amino-acids that range on each side of each cysteine in the primary sequence). Hence, in order to improve automatic methods, it

is worthwhile to wonder which information is actually useful for predicting the disulfide connectivity. Basically, are local environments enough? Some biochemists and structural biologists would answer yes, while some other would definitely argue that the disulfide connectivity is a consequence of the definite proteins folding, thus depending on many other factors: they consider than local environments do not carry any information for disulfide connectivity.

In order to evaluate the impact of local environments of cysteines onto disulfide connectivity, we study an experimental protocol aiming at revealing a potential *affinity* between oxidized cysteines for further bonding. Our hypothesis is that, before the protein folds, some couples of cysteines are more likely to bond than others, given the amino-acids of their primary sequence neighborhood. The best results of past studies were obtained when considering only the 5 ou 6 closest amino-acids of a cysteine in the primary sequence. Our aim is to point out an hypothetic affinity between the local neihborhoods of cysteines. Such an affinity should then be involved in the observed bridges, while being independent from the secondary structure (α-helix and β-sheets). Assuming that such an affinity exists, we suppose that we can detect, extract and evaluate it with machine learning methods launched on observed disulfide connectivity. More generally, we realize this study among several categories of contacts among residues which may driven by some local information.

We thus propose a formalization of the affinity between residues: we focus on a protocol for learning a function representing this affinity from labeled examples available in databases. Starting from machine learning considerations, the main idea of our proposal is to assume that actual bonded residues (positive examples) are not the only examples of high propensity residue pairs: some non-bonded cysteines might also be propitious to form a disulfide bridge according to their neighborhood while some other information does not allow them to actually bond. In previous works, observed bonded residues are considered as positive examples, while non-bonded residues are definitely negative examples: pairs that cannot contact. We argue that our hypothesis may be used for improving usual machine learning methods for predicting residue connectivity. Indeed in a previous work [18], we considered non-bonded residues as unlabeled examples (neither positive nor negative): we then obtained better predictive performances, using a simple naive bayesian classifier, than when non-bonded residues where labeled as negative examples.

In these preliminary works, we considered that there exists an affinity function g, defined on pairs of local environments, which can only take two values: 1 means a high affinity between both environments, while 0 reveals a low affinity. We postulate that pairs with high affinity are more likely to bound than pairs with low affinity. Thus, bounded and unbounded pairs available in proteins databases can be considered as *examples of pairs labeled with* g. Furthermore, these pairs might have been corrupted with classification noise: not all unbonded pairs (resp. bonded pairs) have low (resp. high) affinity. Such a model of noise have already been studied in the machine learning litterature. It is refered to as *class-conditional classification noise* (CCCN), a generalisation of the well-known uniform classification noise (CN). If our base hypothesis is correct, a learning algorithm that is capable of learning from data corrupted by CCCN noise, should also be able to learn the affinity function g from bonded and unbonded pairs of cysteines issued from the proteins databases. Then we should be able to prove that local environments carry some information on the connectivity by checking that pairs with high affinity are more likely to be bonded than pairs with low affinity.

Section 2 is concerned with the presentation and the formal modelling of the biological problem in terms of machine learning methods. Section 3 concerns the algorithms that we propose to learn the affinity function, which are proven to be efficient in some noisy contexts. Section 4 reports some of the numerous experiments we performed: a discussion on the presented results is worthwhile and we hope it will give rise to advices from the whole community of structural biologists and computer scientists.

2 Affinity of protein distant interactions

2.1 Disulfide bridges and salt bridges

A protein may be represented by its primary structure – a sequence of amino-acids– from which a tridimensional structure is gathered. Theoretically, a protein could have many conformations. However, one structure seems to be privileged in a given biological context: the active form of the protein [1]. Nowadays, some interactions are known that contribute to protein stability, such as: hydrogen links, electrostatic interactions, covalent bonds, Wan Der Waals forces. As a matter of fact, the prediction of such interactions should be of great help for the prediction of the structure from the sequence. We are particularly interested by the prediction of affinity between cysteines to bond, making up disulfide bridges. However, the protocol we study can be applied on other contacts such as salt bridges.

Disulfide bridges are involved in the 3D conformation of a protein as covalent bonds between two oxidized cysteines (amino-acid C). Such a physical interaction between two residues is a strong, well-conserved link, thus a strong constraint for the stability of the protein structure. Experimental ways of determining them on proteins, through RMN, X-ray crystallography or site-directed mutagenesis, is a long and expensive process.

Salt bridges are relatively weak ionic hydrogen bonds made up of the interaction between two charged residues. As disulfide bonds, they contribute to the stability of the structure. They involve two residues of the proteins, so are of great interest in our protocol since we study the affinity of two residues for interacting, based on their neighborhood.

2.2 A model of affinity between residues

We present a model and a protocol to answer the question of a local affinity implied in the formation of local interactions between two distant residues of a protein primary structure. We present the protocol through disulfide bonds, but other bonds are also directly concerned as long as the distant contacts involve few residues.

2.2.1 Modeling the data

The primary structure of a protein p can be considered as a word w of Σ^* where Σ represents the set of twenty amino acids or any other similar alphabet. Let $\mathcal{P} \subset \Sigma^*$ be the set of proteins containing an even number of cysteines involved in disulfide bridges (oxidized cysteines). Let $\mathcal{P}_l \subset \mathcal{P}$ be the proteins with $2l$ cysteines involved in disulfide bridges.

Let $\mathcal{G}$ be the set of non-oriented graphs where nodes have degree 1. For a protein $p \in \mathcal{P}$, nodes of the associate graph in $\mathcal{G}$ represent oxidized cysteines of p, and an edge represent a disulfide bond between two cysteines of p. Let $\phi : \mathcal{P} \rightarrow \mathcal{G}$ be a function which associates a graph in $\mathcal{G}$ (the disulfide connectivity) to a protein in $\mathcal{P}$. Then, our aim is to approximate the function ϕ with the highest precision, using examples issued from experiments.

To do so, many authors use local environment of cysteines, *i.e.* amino acids located around the cysteines (figure 1). Usually, segments centered on cysteines of size $2r+1$ are considered. Let P be a probability distribution over $\mathcal{P}$ and let $\Omega_r = \Sigma^{2r+1}$ be the set of protein segments of size $2r+1$. The elements of Ω_r are local environments of cysteines, also called *windows*: a sequence of residues whose center is an oxidized cysteine. On the figure 1, the local environments are $w_1 = LPOCMTN$, $w_2 = DNHCEIS$, $w_3 = EQACPHI$, $w_4 = DSRCNTE$. For $w, w' \in \Omega_r$, let:

- $P(w)$ be the probability that w is a local environment of a cysteine into a protein $p \in \mathcal{P}$
- $P(w, w')$ be the probability that w and w' are distinct local environments of a cysteine into a protein $p \in \mathcal{P}$
- $P(w, w'|l)$ be the probability that w and w' are distinct local environments of a cysteine into a protein $p \in \mathcal{P}_l$
- $P(B(w, w')|w, w', l)$ be the probability that w and w' are bonded knowing that there are distinct local environments of cysteines into a protein $p \in \mathcal{P}_l$.

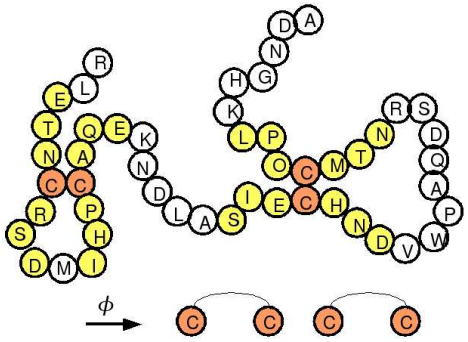

Figure 1. A piece of a sequence of amino-acids, with two disulfide bonds between pairs of oxidized cysteines. Predicting disulfide bonds could be achieved through the function ϕ that computes the correct disulfide connectivity of a protein. There are four local environments, one for each cysteine; here, each local environment is of size 7 (3 amino-acids each side, and the cysteine itself).

2.2.2 Modeling the impact of local environment on bonds

As pointed out in the introduction, past results of automatic methods for the prediction of disulfide bridges based on the proteins sequence are not satisfactory. The error rate remains high (about 40%) while the results are not stable (removing few proteins that are known to be hard to model seems to boost the overall process towards better efficiency). Most of these methods relies upon the local environments of cysteines, namely the environments w modeled above. The aim of our work is to answer the question: *is the local information involved in the formation of disulfide bonds*? Is there any information in the closest neighborhood of the cysteines that would help to predict their bonding during the protein folding process?

In order to answer that question, an affinity measure among cysteines based on their local environment must be highlighted through a functional represesention. The affinity between cysteines must be considered as a necessary, but not sufficient, condition for their actual physical distant interactions. We assume that if such a function exists, then there is a way to learn it from examples. In this section, we draw a model of affinity as well as a protocol to learn it from known disulfide bridges.

Let p be a protein with l bridges ($2l$ involved cysteines). If there is no local information for pairing cysteines into bridges, then there is $2l-1$ pairing possibilities for each cysteine, so $P(B(w, w')|w, w', l) = \frac{1}{2l-1}$. Reciprocally if $P(B(w, w')|w, w', l) = \frac{1}{2l-1}$, there is no local information since the actual pairing does not depend on w nor on w'.

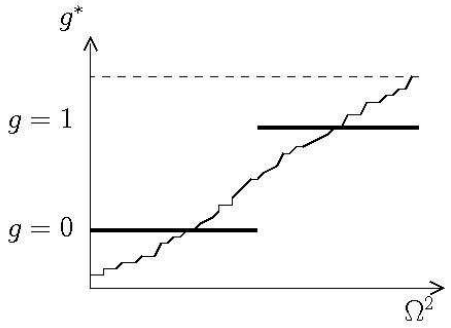

Figure 2. Representation of a two-levels affinity function g in function of the values of $g^* = P(B(w, w')|w, w', l)$. The pairs (w, w') of local environments are ordered by the value of g^*.

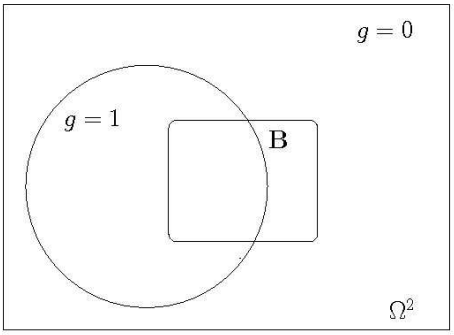

Figure 3. Schematic repartition of local environments pairs for both values of g. Bonded pairs are grouped in set B.

Such an equivalence provides us a probabilistic way to determine if the local context of oxidized cysteines is involved into the formation of the bridges, which requires the estimation of $P(B(w, w')|w, w', l)$. However, estimating $P(B(w, w')|w, w', l)$ without additional hypotheses is impossible. Indeed, with $r = 3$ (which means that we only consider 3 amino-acids on each side of the cysteine in the sequence), the solution space is of size $|\{(w, w'), w, w' \in \Omega_r\}| = 20^{12} \simeq 4.10^{15}$, while only few hundreds examples are available in databases!

Our solution is to assume the existence of an affinity function $g : \Omega_r$ x $\Omega_r \rightarrow Y$ such as:
$g(w_1, w_2) = g(w'_1, w'_2) \Rightarrow$
$P(B(w_1, w_2)|w_1, w_2, l) = P(B(w'_1, w'_2)|w'_1, w'_2, l)$
and
$y < y' \Rightarrow$
$P(B(w_1, w_2)|g(w_1, w_2) = y) <$
$P(B(w'_1, w'_2)|g(w'_1, w'_2) = y')$ $(y, y' \in Y)$.

The simplest situation is $Y = \{0, 1\}$ (Figure 2 : 0 means low affinity between local environments, whereas 1 means a high affinity). In such a case, pairs of windows are partitioned into two classes, corresponding to two affinity levels and $P(B(w, w')|w, w', l) = P(B(w, w')|g(w, w'), l) =$

$$\begin{cases} \alpha_1^l \; if \; g(w, w') = 1 \\ \alpha_0^l \; if \; g(w, w') = 0 \end{cases}$$

Assuming that g exists and plays a role in the interaction, then we must have α_1^l significantly higher than α_0^l. In other words, there are more bonded cysteines when there is a high affinity between their local environments w and w' ($g(w, w')$=1) than when there is a low affinity ($g(w, w')$=0).

The observed bonded and unbonded pairs of local environments centered on oxidized cysteines are then indirect information on g since our model does not exclude that pairs with a high affinity level could be unbonded, neither that a bridge can hold among a pair of cysteines for which g=0.

2.2.3 Bonded and unbonded cysteines as noisy observations of g

From a machine learning point of view, it is an interesting learning context because the observed classes (bonded or non-bonded) of observed examples (local environments of cysteines) do not carry direct information about the affinity function g. This is pictured on figure 3: some pairs of environment are labeled "bonded" while their affinity is 1, and vice-versa. Such an phenomenon is quite usual in machine learning: we interpret these mislabeled pairs as *noisy* labels with regards to the function g to learn. More precisely, using the previous expression of $P(B(w, w')|g(w, w'), l)$, one can observe that pairs such that $g = 1$ correspond to the observation of a bridge with noise $\eta^+ = 1 - \alpha_1^l$, and, symetrically, pairs such that $g = 0$ correspond to a non-bonded pairs with noise $\eta^- = \alpha_0^l$. One could observe that the noise is somehow a measure of mislabelling rates.

On the one hand, this kind of noise is a generalization of the well-known *uniform classification noise* (CN) since it does not make the assumption that positive and negative examples are corrupted by the same noise process (noise rates η^+ and η^- can be different). On the other hand, it is a particular case of *constant-partition classification noise* (CPCN) because there is a fixed number of classes in the space of attributes descriptions to separate (namely two classes: $g = 0$ and $g = 1$). Such a noise has been studied in [23, 10].

2.2.4 Setting up the protocol to learn g

If a local information exists (*i.e.* if local environments of cysteines contribute to their pairing), and if it can be represented by a function in a concept class learnable in CCCN context, then we should be able to detect, extract and evaluate it. Our study thus concerns the setting up, and the test, of a protocol for learning the hypothetic affinity function from proteins where disulfide bridges are known.

In [10], we show that naive Bayes classifiers are identifiable from examples corrupted by a CCCN noise, and we design an efficient algorithm for learning product distributions in this context. Nevertheless, this algorithm assumes the independence of attributes describing data, which does not hold in the case of protein folding. In [23] CCCN-learnable concept classes are proved to be CN-learnable concept classes, in the PAC learning framework. Such a result is of major importance for the local affinity detection protocol we propose: it allows us to search in all CN-learnable concept classes in the PAC framework.

In the next section, we propose an efficient algorithm to learn one of these concept classes: *the linear threshold functions*. This algorithm is a generalization of the perceptron algorithm. [5, 4] sketched two adaptations of it in the CN learning context: we generalize the work of [4] in the CCCN context. We thus propose a perceptron in order to apply our protocol on real datasets for disulfide and salt bridges (see Section 4).

3 A CCCN-learning adaptation of the perceptron Algorithm

The previous section reveals that if an affinity exists between local environments of cysteines, and if it can be represented by a function in a concept class learnable with class-conditional classification noise (CCCN), then it can be learned from examples issued from databases as PDB. In [23], we prove that the concept classes learnable with uniform classification noise (CN) are the same than the concept classes learnable from CCCN noise. As a consequence, *linear threshold functions* are learnable in CCCN context.

In order to experiment the protocol proposed (section 2), this section shows that the perceptron algorithm can be adapted for data corrupted by a CCCN noise. This algorithm generalizes the algorithms proposed in [5, 4] since it can deal with data corrupted by CCCN noise therefore CN.

3.1 Linear Threshold Functions

Let $S = \{(x_1, l(x_1)), ..., (x_n, l(x_n))\}$ be a set of labeled examples[1] (for instance, the set of all pairs of local environments of cysteines), such that $x_i \in \mathbb{R}^m$ (m the size of the descriptive attributes space) and $l(x_i) \in \{-1, 1\}$. S is linearly separable by an hyperplane H if the positive and negative examples of S are separated by H. We refer to a *linearly separable set* S if S is separable by an hyperplane which passes by the origin[2]. Such a hyperplane H is identified by a vector $w^* \in \mathbb{R}^m$ such that $||w^*|| = 1$ and $\forall x \in \mathbb{R}^m$, $x \in H$ if and only if $x \cdot w^* = 0$. We say that w^* separates examples in S with margin σ if $min_{x \in S} |cos(w^*, x)| = min_{x \in S} |w^* \cdot \bar{x}| = min_{x \in S}\ w^* \cdot l(x)\bar{x} = \sigma$ ($\bar{x} = \frac{x}{||x||}$).

In this context, our first purpose is to infer, from a linearly separable set S, an hypothesis w such that w separates positive and negative examples of S: $\forall (x, 1) \in S,\ w \cdot x > 0$ and $\forall (x, -1) \in S, w \cdot x < 0$, or $\forall (x, l(x)) \in S, w \cdot l(x)x > 0$.

Algorithm 1 Sketch of the perceptron algorithm

Require: $S = \{(x_1, l(x_1)), (x_2, l(x_2)), ..., (x_n, l(x_n))\}$

$w = \overrightarrow{0}$

while $\exists (x, l(x)) \in S$ such that $w \cdot l(x)x < 0$ **do**

Let x_{upd} be a vector such that $w^* \cdot x_{upd} > 0$ and $w \cdot x_{upd} < 0$

$w = w + x_{upd}$

end while

Ensure: w such that $\forall (x, l(x)) \in S, w \cdot l(x)x > 0$

3.2 Perceptron Algorithm

The perceptron algorithm [24] is an iterative method for infering a hyperplane w passing by origin, that separates a linearly separable dataset S. A sketch of this algorithm is given on algorithm 1.

In the usual form, $x_{upd} = l(x_B)\bar{x}_B$, where x_B is an example wrongly classified by the current hyperplane w, and $\bar{x}_B = \frac{x_B}{||x_B||}$. This algorithm requires at most $\frac{1}{\sigma^2}$ iterations, where σ is the maximal margin among hyperplanes separating S [4]. It is exponential in the worst case, yet rather efficient in most real problems.

Others possibilities exist for choosing x_{upd}, for instance $\sum l(x_B)x_B$, or any other colinear vector such as the average of the misclassified examples, or its normalized sum. The convergence property holds whenever the selected x_{upd} is wrongly classified by w and not too close from w^*: $cos(w^* \cdot x_{upd}) \geq \sigma$. Let us notice that, if $\forall x \in S$, $|cos(w^*, x)| \geq \sigma$, then $cos(w^*, \sum l(x_B)x_B) \geq \sigma$).

[1] We use annotations 1 or $+$ for bonded pairs – positive examples –, and either -1 or $-$ for unbonded pairs – negative examples –.

[2] One may transform any set S, linearly separable, by an hyperplane H that does not cross the origin, into another set S' that is linearly separable by an hyperplane H' crossing the origin.

3.3 Classification noise

Observed classes of the examples may be corrupted by a noise process: the assigned class for some examples may be wrong, for any reason. The standard perceptron algorithm can not be applied, for it is no more possible to know whether an example is misclassified by w. The most studied noise process is the uniform classification noise (CN), where the labels of examples are supposed to be independently flipped with a constant noise rate $\eta < 0.5$. Two adaptations of the perceptron algorithm in a CN context have been proposed in [5, 4]. The second one presents a direct analysis which computes, when η is known, an estimated value of $\sum l(x_B)x_B$ where x_B are misclassified examples[3]. In order to select a good hypothesis when the noise rate is unknown, it is usual to scan the rate within $[0, 0.5[$ for selecting the hypothesis that leads to the smallest error.

We generalize this noise process by assuming that the noise rate over positive examples is not the same than the noise rate over negative examples, *i.e.* η^+ and $\eta^- \in [0, 1]$, $\eta^+ + \eta^- < 1$ to avoid any ambiguity. Introduced in [23], this new kind of noise was refered to as *class-conditional classification noise* (CCCN). The next section briefly presents an adaptation of the perceptron algorithm within the CCCN-learning framework.

3.4 CCCN-Perceptron Algorithm

This section briefly presents an adaptation of the perceptron algorithm when data is known to be corrupted by CCCN noise, which generalizes the perceptron of [4]. Only the differences with [4] are reported, yet proofs are drastically shortened in order to stay in the scope of this paper. For more details, the reader is adviced to refer to [4, 23].

3.4.1 Known noise rates: η^+ and η^-

Let $S = \{(x, l(x)) \mid l(x) = sign(w^* \cdot x)\}$ be a non-noisy learning dataset. The hyperplane w^* splits S in two sets S^+ and S^-. S^+ is the set of positive examples of S ($w^* \cdot x > 0$), and S^- is the set of negative examples of S ($w^* \cdot x < 0$). In the same way, at each step of the iteration, the current perceptron hyperplane w splits S in two sets S_+ and S_-. That second partition of S is the only one that is observed during the process. When examples of S are corrupted by CCCN noise, we can then estimate the sum of the examples of each of the four parts depicted on figure 4:

$$Sum[S^\alpha_\beta](\alpha, \beta \in \{+, -\}, S^\alpha_\beta = S^\alpha \cap S_\beta)$$

Then, we can estimate $\sum l(x_B)x_B = Sum[S^+_-] - Sum[S^-_+]$.

We prove now that S_+, η^+ and η^- together provide a consistent estimator of both $Sum[S^+_+]$ and $Sum[S^-_+]$. For the same reasons, S_-, η^+ and η^- are proved to establish an estimator of $Sum[S^+_-]$ and $Sum[S^-_-]$. Let us note $Pos[S]$ (resp. $Neg[S]$) the examples of S that are observed positive – bonded pairs – (resp. negative). Then we have:

$$\begin{cases} Sum[Pos[S_+]] = Sum[Pos[S^+_+]] + Sum[Pos[S^-_+]] \\ Sum[Neg[S_+]] = Sum[Neg[S^+_+]] + Sum[Neg[S^-_+]] \end{cases}$$

S^+_+ (resp. S^-_+) only contain true positive examples (resp. true negative examples). The assumption of a random noise process can be written:

- $E[Sum[Pos[S^+_+]] = Sum[S^+_+] \cdot (1 - \eta^+)$
- $E[Sum[Neg[S^+_+]] = Sum[S^+_+] \cdot \eta^+$
- $E[Sum[Pos[S^-_+]] = Sum[S^-_+] \cdot \eta^-$
- $E[Sum[Neg[S^-_+]] = Sum[S^-_+] \cdot (1 - \eta^-)$

where esperances $E[.]$ are considered relatively to the observations we could have of S with the same CCCN noisy process. Using $\widehat{Sum}[]$ to denote the estimations:

$$\begin{cases} Sum[Pos[S_+]] = \widehat{Sum}[S^+_+](1 - \eta^+) + \widehat{Sum}[S^-_+]\eta^- \\ Sum[Neg[S_+]] = \widehat{Sum}[S^+_+]\eta^+ + \widehat{Sum}[S^-_+](1 - \eta^-) \end{cases}$$

After inversion of the system, and assuming $\eta^+ + \eta^- \neq 1$, we obtain:

$$\begin{cases} \widehat{Sum}[S^+_+] = \frac{(1-\eta^-)\cdot Sum[Pos[S_+]] - \eta^- \cdot Sum[Neg[S_+]]}{1-\eta^+-\eta^-} \\ \widehat{Sum}[S^-_+] = \frac{(1-\eta^+)\cdot Sum[Neg[S_+]] - \eta^+ \cdot Sum[Pos[S_+]]}{1-\eta^+-\eta^-} \end{cases}$$

With the same demonstration, we obtain:

$$\begin{cases} \widehat{Sum}[S^-_-] = \frac{(1-\eta^+)\cdot Sum[Neg[S_-]] - \eta^+ \cdot Sum[Pos[S_-]]}{1-\eta^+-\eta^-} \\ \widehat{Sum}[S^+_-] = \frac{(1-\eta^-)\cdot Sum[Pos[S_-]] - \eta^- \cdot Sum[Neg[S_-]]}{1-\eta^+-\eta^-} \end{cases}$$

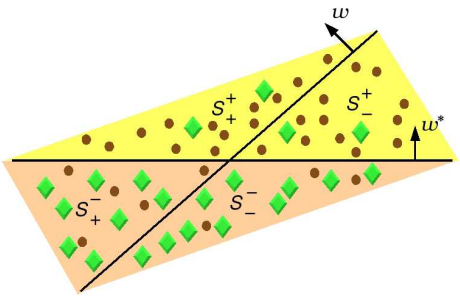

Figure 4. Planar division of cysteines local environments pairs performed by w^*. Pairs having a high level of affinity are above w^* and pairs having a low affinity are below w^*. Current perceptron hyperplane w is indicated to show the four parts of space defined by w and w^*. Points are bonded pairs of cysteines with their local environments while diamonds are non-bonded pairs.

[3] $l(x_B)$ is the correct label of x_B.

Algorithm 2 Perceptron CCCN algorithm (η^+ and η^- known)

Require: $S^\eta = \{(x_1, l^\eta(x_1)), ..., (x_n, l^\eta(x_n))\}$, η^+, η^-, σ
$w = \overrightarrow{0}$, $i = 0$
while $i < \frac{1}{\sigma^2}$ **do**
Compute x_{upd} with current hyperplan w (see formula Section 3.4.1)
$w = w + \frac{x_{upd}}{||x_{upd}||}$, $i \leftarrow i + 1$
end while
Ensure: w such as $w \cdot l(x)x > 0 \; \forall x \in S^\eta$ with high probability

With $x_{upd} = \widehat{Sum}[S_-^+] - \widehat{Sum}[S_+^-]$, we finally compute an estimation of the vector $E[x_{upd}] = \sum l(x_B)x_B$, which only depends on the observations, η^+ and η^-. We thus propose the algorithm 2 as an adaptation of the perceptron algorithm when S is corrupted with CCCN and when the noise rates are known.

3.4.2 Noise rates η^+ and η^- unknown

When noise rates η^+ and η^- are unknown, it is necessary to scan the interval $[0, 1]$ for the values of η^+ and η^- with a step $s \geq \frac{1}{n}$ where $n = |S|$: algorithm 2 is launched for each pair value of the noise rate for computing an hypothesis. However, the principle of empirical risk minimization does not necessarily hold in CCCN context (see [10]).

We thus propose a consistent criterion to select an hyperplane when data is corrupted by a noise CCCN (proof is not given here). This criterion is based on two other criteria independently inconsistent, but which can be laid out to build a consistent whole criterion. The first criterion selects the noise rate associated with the hypothesis w computed within algorithm 2, that minimizes the norm of x_{upd} (computed with w)[4]. The second criterion selects the noise rate that minimizes the sum of the differences between their values (given to the algorithm) and the value of the same parameters computed on S with the hypothesis w within algorithm 2. The main idea of this criterion is that the algorithm must compute, with the real noise rates in input, an hypothesis w for which the computed values of noise rates on S are close to the real values.

Separately, criteria are inconsistent. However we have shown that the only hyperplane that minimizes both criteria is w^*. Then, looking for the hyperplanes that minimize both criteria and selecting the first appearing in these hyperplanes, is a consistent way for selecting a fair hypothesis.

[4] x_{upd} represents an estimate of the sum of examples that are misclassified by hyperplane w.

	Nb of Proteins	Total nb. of bridges
2 disulfide bridges	211	422
3 disulfide bridges	219	657
4 disulfide bridges	88	352
5 disulfide bridges	49	245

Table 1. Statistics of SPX (disulfide bridges)

	Nb of Proteins	Total nb. of bridges
2 salt bridges	182	364
3 salt bridges	166	498
4 salt bridges	136	544
5 salt bridges	86	430

Table 2. Statistics of G3D (salt bridges)

4 Experimentations

We tested the protocol presented in section 2 on two datasets featuring disulfide and salt bridges. We applied the algorithm 2 (section 3) in order to learn a local affinity function supposed to be involved in the pairing of residues.

4.1 Protocol

4.1.1 Datasets

We ran the protocol over two proteins datasets featuring proteins which contains from two to five bonds:

1. the first dataset contains experimentally observed disulfide bridges in proteins; it is known as SPX [2, 9], featuring 1676 disulfide bridges within 567 proteins (Table 1). The homology rate of proteins of SPX is smaller than 30%.

2. The second dataset compiles 1836 intern salt bridges observed in 570 proteins (Table 2); we call it G3D, for it was created from PDB [3] by the ACI GENOTO3D consortium, a french group working on the prediction of the 3D structure of proteins. The homology rate of proteins of G3D is smaller than 25%.

For both kinds of bonds, we distinguish the study according to the number of bonds, since the noise rates induced by proteins containing k bridges are different from the noise rates induced by proteins containing $l \neq k$ bridges (section 2.2.3). Indeed, $l(2l-1)$ pairs of cysteines could be formed within a protein containing $2l$ oxidized cysteines, but only l pairs are actually bonded while $2l(l-1)$ remain unbonded.

4.1.2 Coding of local environments pairs

From a protein p_l containing l bonds ($2l$ oxidized cysteines), we extract $l(2l-1)$ pairs of local environments centered on a cysteine, with radius r (*i.e.* windows of size $2r+1$). We set r to 6 (*i.e.* local environments of size 13, including the central cysteine, because it corresponds to the

Bonds	P(B)
2	0.3333
3	0.2000
4	0.1429
5	0.1111

Table 3. Probabilities to observe a bonded pair in a protein containing $2n$ ($2 \leq n \leq 5$) bonded residues = $1/(2n-1)$

Bonds	P(g=1)	P(B|g=1)	P(B|g=0)
2	0.622 *± 0.088*	**0.338** *± 0.009*	**0.325** *± 0.018*
3	0.436 *± 0.031*	**0.228** *± 0.005*	**0.179** *± 0.002*
4	0.608 *± 0.087*	**0.154** *± 0.003*	**0.124** *± 0.008*
5	0.528 *± 0.051*	**0.116** *± 0.005*	**0.105** *± 0.005*

Table 4. Characteristics of the affinity functions g learned by the CCCN-perceptron algorithm on SPX.

Bonds	P(g=1)	P(B|g=1)	P(B|g=0)
2	0.649 *± 0.037*	**0.381** *± 0.009*	**0.246** *± 0.012*
3	0.485 *± 0.068*	**0.243** *± 0.005*	**0.160** *± 0.007*
4	0.482 *± 0.061*	**0.174** *± 0.005*	**0.114** *± 0.003*
5	0.593 *± 0.047*	**0.129** *± 0.003*	**0.084** *± 0.005*

Table 5. Characteristics of the affinity functions g learned by the CCCN-perceptron algorithm on G3D.

best results we and other authors have obtained so far). For any pair (w_i, w_j) of p_l local environments, we extract 169 residue pairs (A_i, A_j) ($i, j \in \{1, ..., 13\}$) where $A_i \in w_i$ and $A_j \in w_j$. Each pair (w_i, w_j) is modeled with a vector of $\mathbb{R}^m$, where m is the number of ordered pairs of residues within the alphabet Σ, and where each coordinate is the number of times a pair is observed in (w_i, w_j). The alphabet Σ contains a symbol for each amino-acid, and a symbol X which denotes any unknown amino-acid ($|\Sigma| = 21$ and $m = 231$) . The coding of salt bridges is quite the same, except the central amino acid since salt bridges occur between two charged residues (Aspartic Acid (D) or Glutamic Acid (E) with Lysine (K), Arginine (R) or Histidine (H)).

4.2 Results

We launched two experiments: one for learning the affinity function involved in disulfide bridges (table 4 and figure 5), and the other for learning the affinity function involved in the salt bridges formation (table 5 and figure 6). Three

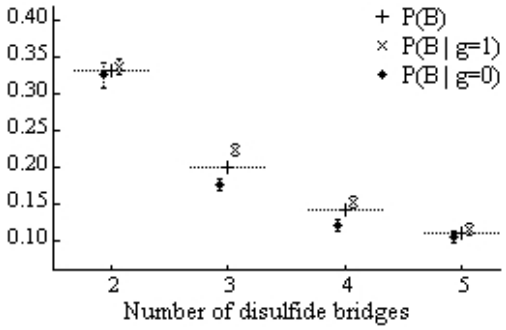

Figure 5. Graphical view of table 4.

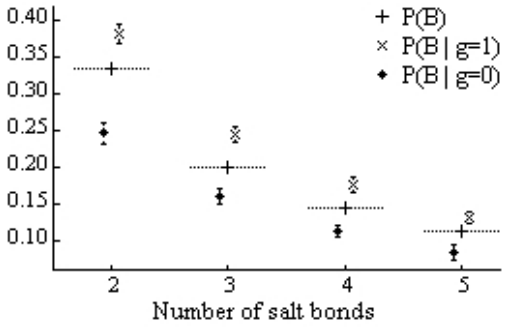

Figure 6. Graphical view of table 5.

criteria are reported with standard square deviations:

1. $P(g{=}1)$, the probabilities that pairs of local environments have a high level of affinity (computed with linear functions infered with the perceptron algorithm),
2. $P(B|g{=}1)$, the probability to observe a bond knowing that the pair is predicted to have a high level of affinity,
3. $P(B|g{=}0)$ the probability to observe a bond knowing that the pair is predicted to have a low level of affinity.

Each reported result is an average of five 10-fold cross-validations on the subset of proteins containing n bridges ($2 \leq n \leq 5$). Standard deviations are italized.

It is worthwhile to notice that, in order to ensure that the detected local information is not correlated with beta sheets or alpha helix in proteins, we launched experiments to look after (i) the ratio of residues involved both in bridges and beta sheets, (ii) the ratio of residues involved both in bridges and alpha helix, and (iii) those only involved in either disulfide or salt bridges. These experiments show that no correlation exists between the level of affinity predicted for local environments pairs and these 2D structural elements.

As discussed in section 4.3, we observe that results on salt bridges show that an affinity between local environments is learnt, whereas there is no evidence that such an affinity exists that would guide the formation of disulfide bridges.

4.3 Discussion

On one hand, results on salt bridges reveal that a clear signal is detected: there exists local information that is involved in the formation of salt bridges. That signal is quite constant along the experiments (standard deviations are small in comparison with the differences of probabilities $P(B|g = 1)$ and $P(B|g = 0)$). Figure 5 shows that whatever the number of bonds is, our algorithm learns an affinity function g that always classify more observed bonds as having high affinity than having low affinity. In other words, we pointed out an affinity function between local environments of salt bridges. The detected affinities might be explained either by the ionic nature of salt bridges, which often involves the charge of their local environments, or/and by the hydrophilic property of many residues around salt bridges. Thus, the results that we obtained seem to validate our protocol for detecting an affinity function between local environments of paired residues.

On the other hand, the results on disulfide bridges pictured on figure 4, are not as clear as expected: probabilities $P(B|g = 1)$ and $P(B|g = 0)$ are really close to the baseline probabilities given in table 3 (especially for proteins containing 2 and 5 bridges). These results may be explained by several independent reasons.

1. **Biology reality**. The first insight of our results is that there might be no local information that would guide the formation of disulfide bridges during the 3D conformation of proteins. Such an explanation would be shared with many biologists and biochemists: disulfide bonds are so strong interactions (covalent links) that the propensity between their environments is not enough determining for guiding actual bonds.

2. **Data sparsity**. The perceptron algorithm is known to fail on high dimensional spaces, as here where vectors are of size 231 and many values are null, because it does not optimize the margin between the infered hyperplane and the examples. In order to estimate the impact of sparsity on our experiments, we used hyperplanes infered by this algorithm, for relabelling the learning data. A soft-margin algorithm has then been launched on these re-labelled dataset in order to optimize the margin. However, no significant improvement has been observed on test data: the sparsity seems to be not responsible of the flat results.

3. **Learning a function in the wrong concept classes**. The affinity function g that we try to learn might be not representable by a linear threshold function such as learnt by any perceptron. It might be a function of another concept class: obviously, we still have to design other algorithms adapted to CCCN noise in other concept classes.

However, this work does not allow us to known which assumption is the most probable. The case of the disulfide bridges stays an open question. In future work, we shall have to learn the affinity function g in other concept classes to check if the results are similar to those reported here.

4.4 Conclusion

These experiments validate our protocol and show that this method make possible to detect a true signal of affinity between local environments, with a good quality. Results obtained on the salt bridges (G3D) are quite relevant because they confirm that the perceptron algorithm with CCCN noise always outputs an affinity: bonded pairs are mostly classified as having a high level of affinity. The stability of the results confirms the existence of an affinity function. Consequently, we can conclude that the local environment of charged residues is involved in the formation of salt bridges.

The perceptron algorithm might not be the right one for detecting affinities of environments in forming disulfide bridges, assuming that such a phenomenon exists and is implied in the pairing of oxidized cysteines. As a consequence, we can not currently be sure that there is no local affinity contributing to the pairing of oxidized cysteines: the chosen algorithm did not detect them. Thus, other algorithm must be tested within our CCCN noise-based protocol.

5 Conclusions and future works

We presented a machine-learning based protocol to answer the question of the presence of local affinities that would be involved in the pairing of distant residues in proteins. We validated this protocol since we were able to learn an affinity between local environments of salt bridges. However, the same protocol has not yet indicated any impact of local environments on the formation of disulfide bonds. More generally, the protocol can be used in order to detect any affinity between pairs of local environments residues.

In a machine learning point of view, this work is a success for it proves that it is crucial to theoretically study other algorithms fitting the CCCN context. We expect finding an algorithm for learning another concept class, that would be adapted to the highlighting of an affinity between local environments of residues to form bonds.

The presented protocol initiates the state of the art for the question of the existence of local affinities involved in local interactions. Yet many studies have to be done using this protocol for surrounding this question. We now hope that several biologists and computer scientists will help us to prove an answer, whatever it could be.

References

[1] C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, jul 1973.

[2] P. Baldi, J. Cheng, and A. Vullo. Large-scale prediction of disulphide bond connectivity. In *Proceedings of NIPS' 05, Advances in Neural Information Processing Systems*, pages 97–104, 2005.

[3] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[4] A. Blum, A. M. Frieze, R. Kannan, and S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. In *IEEE Symposium on Foundations of Computer Science*, pages 330–338, 1996.

[5] Bylander. Learning linear threshold functions in the presence of classification noise. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1994.

[6] A. Ceroni, P. Frasconi, A. Passerini, and A. Vullo. Predicting the disulfide bonding state of cysteines with combinations of kernel machines. *J. of VLSI Signal Processing Systems*, 35(3):287–95, 2003.

[7] A. Ceroni, A. Passerini, A. Vullo, and P. Frasconi. Disulfind: a disulfide bonding state and cysteine connectivity prediction server. *Nucleic Acids Research*, 34(suppl.2):177–181, 2006.

[8] J. Cheng and P. Baldi. Three-stage prediction of protein β-sheets by neural networks, alignments and graph algorithms. *Bioinformatics*, 21(1):75–84, 2005.

[9] J. Cheng, H. Saigo, and P. Baldi. Large-scale prediction of disulphide bridges using kernel methods, two-dimensional recursive neural networks, and weighted graph matching. *Proteins*, 62(3):617–629, 2006.

[10] F. Denis, C. N. Magnan, and L. Ralaivola. Efficient learning of naive bayes classifiers under class-conditional classification noise. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 265–272. ACM Press, 2006.

[11] P. Fariselli and R. Casadio. Prediction of disulfide connectivity in proteins. *Bioinformatics*, 17(10):957–964, 2001.

[12] P. Fariselli, P. L. Martelli, and R. Casadio. A neural network-based method for predicting the disulfide connectivity in proteins. In *Proceedings of KES 2002, Knowledga based intelligent information engineering systems and allied technologies*, 2002.

[13] P. Fariselli, P. Riccobelli, and R. Casadio. Role of evolutionary information in predicting the disulfide-bonding state of cysteine in proteins. *Proteins*, 36(3):340–346, 1999.

[14] F. Ferrè and P. Clote. Disulfide connectivity prediction using secondary structure information and diresidue frequencies. *Bioinformatics*, 21(10):2336–2346, 2005.

[15] A. Fiser and I. Simon. Predicting the oxidation state of cysteines by multiple sequence alignment. *Bioinformatics*, 16(3):251–256, 2000.

[16] P. Frasconi, A. Passerini, and A. Vullo. A two-stage svm architecture for predicting the disulfide bonding state of cysteines. In *Proc. of the IEEE Workshop on Neural Networks for Signal Processing*, 2002.

[17] E. Hutchinson, R. Sessions, J. Thornton, and D. Woolfson. Determinants of strand register in antiparallel β-sheets of proteins. *Protein Science*, 7:2287–2300, 1998.

[18] C. N. Magnan. Asymmetrical semi-supervised learning and prediction of disulfide connectivity in proteins. In *R.I.A. New Methods in Machine Learning: Theory and Applications*, volume 20(6), pages 673–695. Hermes Lavoisier, 2006.

[19] P. L. Martelli, P. Fariselli, L. Malaguti, and R. Casadio. Prediction of the disulfide-bonding state of cysteines in proteins at 88% accuracy. *Protein Science*, 11(11):2735–9, 2002.

[20] P. L. Martelli, P. Fariselli, L. Malaguti, and R. Casadio. Prediction of the disulfide bonding state of cysteines in proteins with hidden neural networks. *Protein Engineering*, 15(12):951–953, 2002.

[21] J. S. Merkel, J. M. Sturtevant, and L. Regan. Sidechain interactions in parallel β-sheets: the energetics of cross-strand pairings. *Structure Fold Description*, 7(11):1333–1343, 1999.

[22] M. Mucchielli-Giorgi, S. Hazout, and P. Tufféry. Predicting the disulfide bonding state of cysteines using protein descriptors. *Proteins: Structure, Function, and Genetics*, 46(3):243–249, 2002.

[23] L. Ralaivola, F. Denis, and C. Magnan. Cn = cpcn. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 721–728, 2006.

[24] F. Rosenblatt. Principles of neurodynamics. In *Spartan Books*, 1962.

[25] J. Ruan, K. Wang, J. Yang, L. A. Kurgan, and K. Cios. Highly accurate and consistent method for prediction of helix and strand content from primary protein sequences. *Artificial Intelligence in Medicine*, 35(1-2):19–35, 2005.

[26] S. Sen. Statistical analysis of pair-wise compatibility of spatially nearest neighbor and adjacent residues in α-helix and β-strands: Application to a minimal model for secondary structure prediction. *Biophysical Chemistry*, 103:35–49, 2003.

[27] J.-N. Song, M.-L. Wang, W.-J. Li, and W.-B. Xu. Prediction of the disulfide-bonding state of cysteines in proteins based on dipeptide composition. *Biochemical and Biophysical Research Communications*, 318(1):142–147, 2004.

[28] R. E. Steward and J. M. Thornton. Prediction of strand pairing in antiparallel and parallel β-sheets using information theory. *Proteins*, 48:178–191, 2002.

[29] A. Vullo and P. Frasconi. A recursive connectionist approach for predicting disulfide connectivity in proteins. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pages 67–71, 2003.

[30] A. Vullo and P. Frasconi. Disulfide connectivity prediction using recursive neural networks and evolutionary information. *Bioinformatics*, 20(5):653–659, 2004.

[31] C.-T. Zhang, Z.-S. Lin, Z. Zhang, and M. Yan. Prediction of the helix/strand content of globular proteins based on their primary sequences. *Protein Eng.*, 11(11):971–979, 1998.

[32] E. Zhao, H.-L. Liu, C.-H. Tsai, H.-K. Tsai, C. hsiung Chan, and C.-Y. Kao. Cysteine separations profiles on protein sequences infer disulfide connectivity. *Bioinformatics*, 21(8):1415–1420, 2005.